AMENDMENTS TO THE SPECIFICATION

Please replace paragraph [1020] with the following amended paragraph:

[1020] Referring to FIG. 2, a portion of Z buffer 118 and a corresponding portion of the hierarchical image depth buffer 122 is shown. The Z buffer 118 includes a plurality of depth values, such as individual depth value ~~[[202]]~~ 203. Each depth value entry in the Z buffer 118 contains a depth value for a particular pixel in a two-dimensional projection of a three-dimensional scene. The depth value identifies a Z coordinate location for a pixel, assuming an X, Y, Z three-dimensional coordinate system. The Z buffer is subdivided into a plurality of sets of depth value entries. In FIG. 2, sets 204, 206, 208, 210, 212, 214, 216, and 218 are illustrated. Each set includes a plurality of individual depth values. In FIG. 2, each set contains sixteen depth values, illustrated as a 4 x 4 array. The hierarchical image depth buffer 122 corresponds to the Z buffer 118. A portion of the hierarchical image depth buffer 122 is shown in FIG. 2. The illustrated portion of the hierarchical image depth buffer 122 includes a first entry 220 and a second entry 224. The first entry 220 corresponds to a set within the Z buffer, such as set 204. The second entry 224 corresponds to a different set within the Z buffer 118, such as set 208. Each entry, such as entries 220 and 224, contains a plurality of data items.

Please replace paragraph [1022] with the following amended paragraph:

[1022] Referring to FIG. 3, an illustrative scene containing two different objects is shown. The illustration 300 includes a viewpoint 302, a viewplane 304, a first object 306, a second object 308, and a background plane 310. The illustration also includes a coordinate system that contains a depth (Z) coordinate [[312]]. The first object 306 contains a first sub-object 316 that is a portion of the first object 306. Similarly, the second object 308 contains sub-objects 312 and 314 that are portions of the second object 308. Each of the sub-objects is triangular in shape. Assuming that a computer graphics application instructed the second object 308 to be rendered, the sub-objects 312 and 314 would be processed by the rendering engine. During a first pass, both the sub-objects 312 and 314 would be rendered onto the image plane. Subsequently, when the first sub-object 316 is rendered, since sub-object 316 (and the pixels within sub-object 316) block sub-object 312 with respect to the viewpoint 302, sub-object 316 would be rendered and sub-object 312 would be occluded. In this case, the rendering engine would determine that each of the pixels forming the sub-object 312 should be overwritten with the pixels from sub-object 316. The illustration of FIG. 3 provides a basis for referencing the pixel-by-pixel determination of visibility or occlusion performed by the visibility determination method handled by visibility detection logic 114. Typical software applications contain more than the two objects illustrated in the simple drawing of FIG. 3, and the many objects in a typical scene may be interwoven and have various complex shapes and sizes.

Please replace paragraph [1079] with the following amended paragraph:

[1079] The example system uses a three level hierarchy for initialization purposes. The first two levels are the depth information hierarchy. The third level consists of an initialization flag for each 16 x 8 "Superblock" of 128 pixels[[.]] <u>such as the block of initialization flags 133 shown in Fig. 1. Thus, a third level hierarchical buffer contains a plurality of initialization flags, each of the initialization flags for a super-block of pixels corresponding to a plurality of the entries within the hierarchical image depth buffer.</u> There are 8 TileZ entries for each Superblock and they occupy 64 bytes. There are 6 K Superblocks and Superblockinit flags for a 1024 x 768 pixel image. Superblockinit flags are 'cached" in a small number of 64 byte buffers in the graphics engine.